

A Comparative Approach for SCRUM and FDD in Agile

Munazza Umbreen¹, Jalil Abbas¹, Syed Muqsit Shaheed²

¹Department of Computer Science & IT Govt. College University Faisalabad (Layyah Campus), Layyah, Pakistan

²Department of Computer Science & IT, University of Lahore, Lahore, Pakistan

Email: sjshah786@gmail.com

Abstract- Rapid development in the contemporary world in the sphere of technology and business, usually the requirements don't seem to be manifest, hence, a consistent trend of change is palpable in the process of development, therefore, these consistent changes in the requirements render the software development all the more cumbersome. On the other hand, use of traditional software development methods like waterfall method is of no good and pragmatic option, because traditional software development methods are inflexible to requirements and software is likely to be late and over budget. For the purpose of development of high quality software which can optimally satisfy the customer, organizations can make use of software development methods, like as Agile methods which are more appropriate and flexible to changing requirements at any stage in the development process. Many of the organizations have begun to employ Agile methodologies. Agile methodologies have been ushered in to cater to the new and emerging day to day requirements of the software development organization. Minor research work has been initiated on Agility of Scrum and Feature Driven Development (FDD). This research boils down to select these methodologies and their comparison for searching out a better and compatible use of the study area. A host of issues connected to FDD and Scrum have been identified. An attempt has been made to derive some solutions to overcome the problems and issues involved. By applying these results organizations can ameliorate their processes, systems and components. The data was gathered by using online questionnaire, discussion, survey and connected literature. The results of these transpired the ways of using SCRUM and FDD either alone or in combination with each others to fulfill the client's requirements and product management.

Keywords: Agile Development process; Software Development Life Cycle (SDLC); SCRUM method; FDD in Agile; traditional software development method.

I. INTRODUCTION

A software development technique in software engineering is a structure that is used to develop, plan and manage the process of developing software product. A development technique also includes the integrated development environments (IDEs), model based on development, computer aided software development etc. There are many software development methodologies and every methodology has its own life cycle known as Software Development Life Cycle (SDLC). SDLC supports a development team to develop effective software product.

Waterfall model is a first process methodology which is linear in nature and was first described by Winston Royce in 1970 [1]. It is still widely in vogue in both small and large projects [2]. The waterfall methodology comprises of five main consecutive phases: requirement, design, implementation, verification and maintenance for completing a sequential software development process as depicted in fig-1. All phases are linear and sequential not incremental that has been used to plan, manage and control all process of developing information system. So the output is developed at the end of the process which may or may not meet the user's demands.

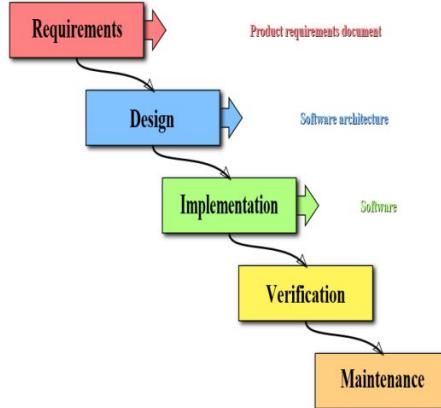


Figure 1: Life Cycle of Waterfall Methodology

The waterfall model progress in well systematic and sequential phases produces valuable, reliable and maintainable small software products defined the resources, tasks, entities and work flow of every phase. Waterfall method was suitable for small and non changing environment development processes as it was not an iterative method. A developer can't go back to remove an error if it was found. That's why it was not considered as a good methodology for complicated and incremental projects. There was a need of such a type of methodology which can manage changes during the processes [3]. The rest of the paper is designed as follows section 2 discusses the Agile software development methodology. Section 3 comprises of an overview of two agile methodologies: SCRUM and FDD. In section 4 we discussed the data collection methodology followed by the analysis of data. Results are elaborated graphically. At the end conclusion is given.

II. AGILE SOFTWARE DEVELOPMENT

Agile software development methodologies have important applications in the area of software project management, software schedule management in software industry. Agile is an incremental and iterative approach for developing software products. Agile software development processes follows the SDLC which consists of the requirements, architecture and design, development and test and feedback. This suggested approach was later known as agile development process which is elaborated as figure 2.

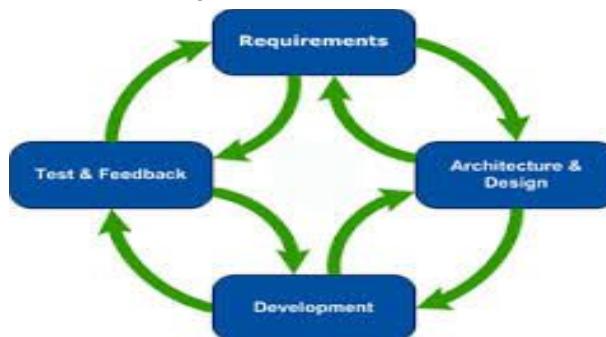


Figure 2: Agile Software Development Process

The difference between traditional methodologies and agile software development methodologies are: traditional methodologies are complicated, constant, sequential, oddest and highly mature level but agile methodologies had been adopted in many software development projects due to its ability to better cope with frequent changes in requirements [4]. The agile methodologies are suitable for the software development processes where upcoming requirements are changed during each process. Keen observation of analysis shows that agile methodologies help us to acquire better outcome with high quality, more client satisfaction, efficiency and management within the specified time and expenses [5]. Agile methodology encourages the developer team to meet with the customers on regular basis in order to validate and verify their requirements. Agile is now the best approach to software development adopted by some organizations in their projects such as yahoo and Google. Agile software development is iterative and incremental method development by self-organization and different team software is developed. Agile is able to cope with change and manage the risk effectively while the traditional approach cannot change during the project. Agile method helps the Manager ability to understand the plan of the organization and ability to change the requirement of programs [6].

Well-known software development methods include: SCRUM, FDD, adaptive software development, agile modeling, agile unified process, crystal methods, disciplined agile delivery and extreme programming etc. SCRUM method is an agile method that focuses on better managing iterative development and best suited for large and complex products. FDD is features based software development process. It focuses on design and code inspection which results in high quality software. In agile methods the requirements are gathered at a regular interval and also the requirements can be changed at any stage in the development phase. In this study we focus on the two given below methodologies of Agile which are SCRUM and FDD.

III. AN OVERVIEW OF SCRUM AND FDD

FDD and SCRUM are two different methodologies of agile. Agile overcomes the main weakness of waterfall methodology by its iterative development procedure. Each iteration is meant to be short (1-2 weeks) and includes all of the following steps; Gathering user requirements, Design and documentation, Development, Testing and Deployment which discovers the design errors are at early stages of development. Before comparing FDD and SCRUM methodologies, a short overview of each practice is described.

A. SCRUM

SCRUM technique is more suitable for short term software development and is based on organization accepted best practices, used and proven for decades. SCRUM is a simple project management framework. SCRUM framework also reduces risk and builds a more transparent and trusting relationship between software development team and customers. SCRUM allows customers to change their mind and evolve their requirements during the development process. SCRUM is simple project management framework ensures that a motivated team is always working on building the customers most important features [7]. SCRUM technique help to manage the project more efficiently instead of designing and developing a project. But this agile methodology is closely similar to iterative engineering and is administered monthly. The SCRUM technique works quite well for developers at organizations and is a big improvement over the traditional method. Developers thought the SCRUM technique helped team members get involved in projects. The SCRUM technique has many detail documents, such as class diagrams, sequence diagrams, activity diagrams, communication diagrams, and use cases were significantly reduced, or disappeared [8].

SCRUM includes a “process framework” that is combination of predefined phases and actions. It is a highly iterative method which develops software product in iterations commonly known as “Sprints”. Each sprint follows all phases of waterfall software development method i-e planning, designing, developing, and testing. Project success mainly depends on the effective communication between the development team and clients during the complete developmental process. Sprint is like a minor development process which gives a functional output at the completion. It takes 2-4 weeks for completion. The time table for sprints’ completion is decided by product owner and the SCRUM team. The SCRUM life cycle starts with maintaining and creating a Product Backlog by product owner with the help of data collected from

the management, development team, business place, customers and the end users. A “PRODUCT BACKLOG” consists of all the customer’s specifications and demands. Sprint backlog is maintained by product backlog during the sprint planning meeting. Sprint backlog can’t be changed by the team members only SCRUM master can do it. At the end of every sprint, a small functional deliverable unit is made by the team. The remaining sprint processes are presented by the burn down charts that leaves over work and time period.

B. FDD

Jeff De Luca and Peter Coad introduced FDD in 1997 and is a lightweight, iterative and incremental software development process. It emphasizes quality and delivers frequent, tangible working results at all steps. FDD also provides accurate and meaningful progress and status information, with the minimum of overhead and disruption for the developers. It blends a number of industry-recognized best practices into a cohesive whole. Discovering list of features is a critical process. The quality of this step largely defines how precise the project will be tracked, how maintainable and extensible the code will be. This process requires full-time participation of customers. The outcome of this step is the UML diagrams of problem domain. If the two-way development tool is used, than UML diagrams are backed up by the compile able code in the target programming language [9].

FDD consists of five high level activities: Domain Object Modeling: It advocates light modeling up front to understand the shape and scope of the application. Developing by Feature: Team builds a feature list. Each feature represents 1-10 days worth of effort. A feature is a small piece of client-valued function .Plan by feature: After the feature list is completed, the next step is to produce the development plan; assigning ownership of features (or feature sets) as classes to programmers. Design By Feature (DBF): set of features are assigned to Feature teams Together with the corresponding class owners, the chief programmer works out detailed sequence diagrams for each feature and refines the overall model. Build By Feature (BBF): The class owners develop the code for their classes. After a unit test and a successful code inspection when chief programmer is satisfied then completed features are promoted to the main build. FDD includes: The principle of least privilege, the principle of failing securely, the principle of securing the weakest link, the principle of defense in depth, the principle of separation of privilege, the principle of economy of mechanism, the principle of least common mechanism and the principle of complete mediation [10].

IV. DATA COLLECTION METHODOLOGY

Data for this study is collected through an anonymous online survey of software development professionals who are most likely to be early adopters of agile development or are currently working in Agile environment. For this purpose we join ten agile forums, Agile Alliance, LinkedIn and yahoo online discussion groups that focuses on agile software development. We remained in touch with these groups and participated in their activities. Finally, we decided to discuss the Agile methodologies in detail. For this purpose we selected SCRUM and FDD for our research. Since thousands of people are the members of these groups so after obtaining permission from the groups’ owners we posted a solicitation message inviting members who had experience using agile development approaches to complete an anonymous web-based survey we also put forward a questionnaire for the comparison of these two methodologies. In this research work, four basic attributes (quality, cost, risk and time) of project development are taken to test the preference of software professionals based on each aspect separately.

A. Analysis of Data

A basic milestone of the current research was to disclose the ‘pros and cons’ of SCRUM and FDD to compare them and make conclusion about the effectiveness of one methodology over other. After the data collection the analysis step is done by this web-based survey many issues related to the ‘problem statement’ were recognized. These issues were so helpful to find out suitable results. The comparative study defined the important factors of one methodology over the other. These factors are:

- Quality
- Cost
- Risk
- time

On the basis of survey and the questionnaire the comparison for critical factors is given in fig-3.

Factors	Quality	Cost	Risk	Time
Waterfall	Predictive	Expensive	High Risk	Time Consuming
SCRUM	More Adaptive	Inexpensive	Reduces Risk	Delivers on Time
FDD	High Quality	Not clear Cost estimation	Lack of Risk Analysis	Possible Time

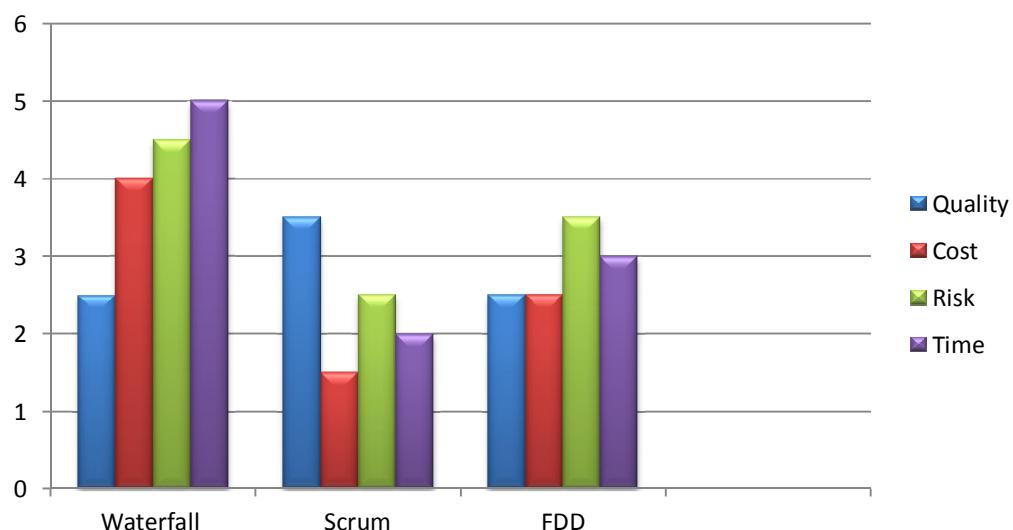


Figure 3: Comparison for Critical Success Factor

Both have some benefits as well as shortcomings with respect to development environment. It is analyzed that both methodologies must be refined by adding and removing some features. Both software development methodologies have their own rules and regulations. By evaluating the responses of the expert developers, it was discovered that it is better to use combinations of good activities of different agile methodologies for developing incremental, secure, reliable, valuable and suitable software product.

B. Product Delivery

Users don't keep interest on the development methodology used by the developers. They are only satisfied by the quick and effective deliveries. SCRUM produces quick output after completion of sprints and integrates their unit functional deliverable to produce the complete software product. FDD gives output only at the end of the process. No unit deliverable is given by FDD during the process.

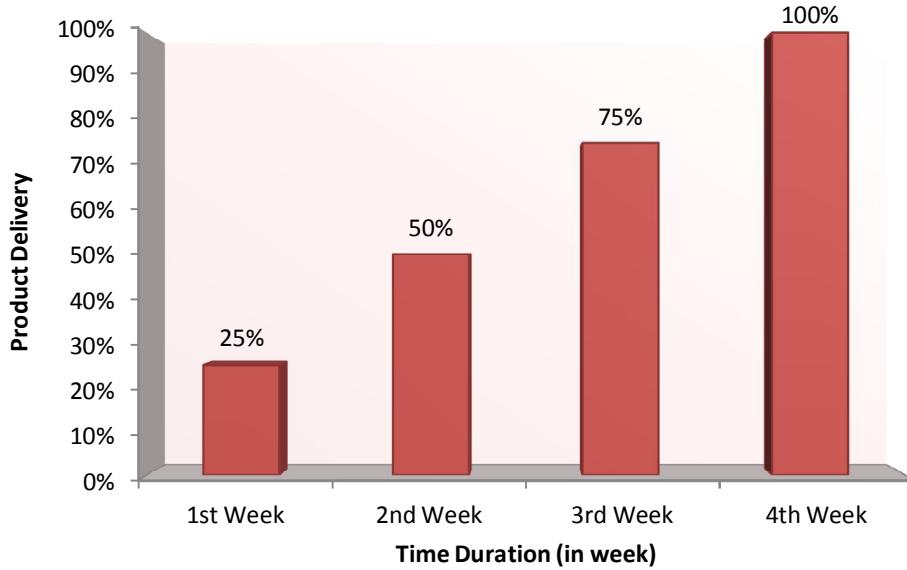


Figure 4: Product delivery in SCRUM

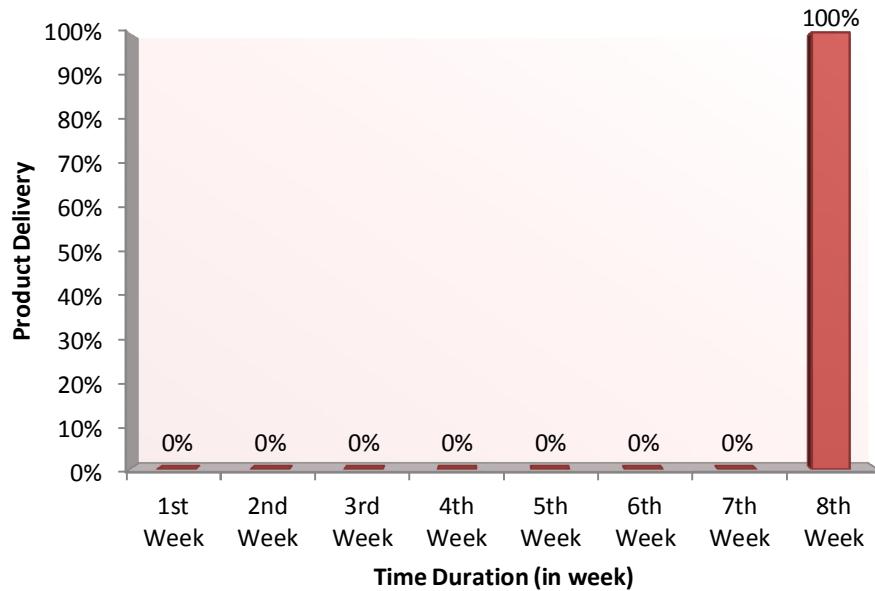


Figure 5: Product delivery in FDD

The figure 4 depicts that SCRUM produces output after each iteration i.e. after one week but FDD gives out put only at the end i.e. 8th week as shown in above figure 5.

C. Comparison of characteristics of FDD and SCRUM

On the basis of our research we can provide a detailed comparison of both methodologies of Agile i.e. SCRUM and FDD. These methodologies differ with each other by the following characteristics as depicted in Table 1:

Table-1: Comparison of Characteristics of FDD and SCRUM

Characteristic	SCRUM	FDD
Human oriented software development environment	It provides such type of opportunities to change behavior of the software product according to human mind and human oriented software development environment.	FDD shows less human orientation as a complete walkthrough is held for the requirements gathering, whereas human mind changes with new ideas continuously.
Simplicity	SCRUM shows more simplicity by making individual functional unit called 'sprints'. Every sprint is itself a minor product and is simply understandable and applicable.	FDD is complex as compare to SCRUM.
Requirement Elicitation	SCRUM collects and describes high level requirement by daily SCRUM, sprint meetings and SCRUM meeting.	FDD requirements are gathered only by a walkthrough.
Pair Programming	It is a development technique to develop a code by two developers for getting good output. SCRUM involves pair programming as the complete development is responsible for the code. No code ownership is a part of SCRUM.	Code ownership is the main element of FDD.
Code Inspection	SCRUM, testing process is performed by its own team member who can't fix his own errors.	Code inspection is better in FDD and a separate tester performs code inspection and fixes errors.
Cross Functionality	It is the main feature of SCRUM. There is no team leader in the SCRUM team but they work collectively. and responsibility of product is on the whole team.	FDD lacks cross functionality and deal with code ownership.
Client's Feedback	It shows maximum client's feedback after every sprint and ultimately required product. Changes are performed during the process according to the client's feedback in SCRUM.	It shows less feedback as it produces output only at the end of development process.
Agility	The term "agility" means the capability of moving and changing speedily according to wish and control. Good in it because roles of SCRUM master, product owner and team members more satisfy to agility.	FDD shows less agility than that of SCRUM because roles of project manager, development manager and chief architecture of FDD methodology are more likely to the conventional development method.
Fewer Errors	SCRUM suits for error handling and focuses on quality by performing these activities because Pair programming and unit testing focus on error removal at the spot.	Bad in FDD because FDD software development process doesn't involve in pair programming and unit testing.
Risk of failure	SCRUM can fully handle the exceptional errors during the entire process and gives quality software product.	FDD performs inspection at the end of feature development and can't fully handle unexpected errors at the spot.

Verification and Validation	SCRUM deeply support verification and validation processes.	FDD gives only verified products but not adequate valid product that can fulfill the new demands.
Collaboration and Communication	SCRUM fully supports face to face communication and gathers changing demands of users during the whole development process.	FDD has limited communication and requirements are gathered only by walkthrough. It can't perform alterations during process according to client's demands.
A comparison of Client's meetings in FDD and SCRUM is depicted in fig-6		
Certification	SCRUM is a highly certified software development process as it requires certified SCRUM masters and developers.	FDD has no such certification. This is the main reason that SCRUM is more popular in software developing industry.
Flexibility	SCRUM is so flexible that changes are done according to client's requirements even in the early steps of development process. Flexibility is required especially for web designing and implementation.	SCRUM shows more flexibility over FDD.
Adaptability	SCRUM is suitable where adaptability requires.	FDD is less adaptable because its first three steps are closely related to traditional software development process. FDD don't warmly welcome to alterations during the process of software development.
Quality Assurance Process	SCRUM is a highly quality assurance process as it involves frequent alterations and deliveries. Sprint testing keeps the process tracked and measured.	Testing process in FDD only performs at the end which fails to assure the product quality in fast software development industry.

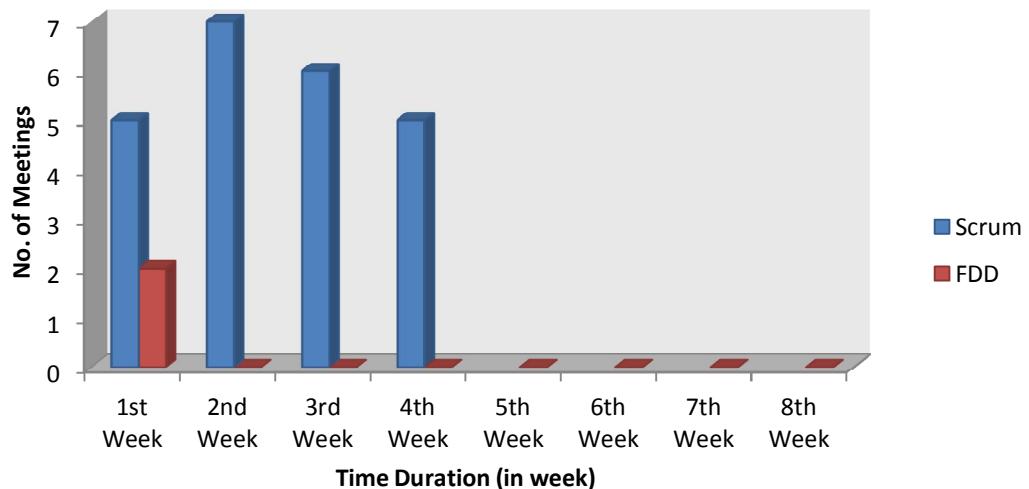


Figure 6: Client's meetings in FDD and SCRUM

V. CONCLUSION

In the recent years, software development organizations are striving to adopt agile software development methods and techniques. Successful agile adoption leads to producing higher quality software, enhances developers moral at a lower cost than the traditional waterfall model approach. This research study sets out to use survey and questionnaire data to explore the comparison between the Agile methodologies of SCRUM and FDD. From the comprehensive analysis it is revealed that SCRUM gives better overview of the project, less bugs, a more focus on quality of the product, manages well the changing priorities and sequence of activities and have a committed team. FDD leads to move to large projects, reduces risk via iteration of design and build in small chunks, it estimates the cost of project by feature which leads to greater accuracy. But FDD is not as powerful on smaller projects, gives no written documentation and high reliance on chief programmer. So we concluded this research that SCRUM is more popular in Agile methodologies.

VI. ACKNOWLEDGEMENT

I, personally feel extremely beholden to Prof (R) Ghulam Qasim Shah who with self-abnegation graciously spared his time and reviewed this paper.

REFERENCES

- [1] ROYCE, Winston W. Managing the development of large software systems. In : *proceedings of IEEE WESCON*. 1970. p. 328-388.
- [2] ALSULTANNY, Y.A., WOHAISHI, A.M. Requirements for Software Quality Assurance Model, *Second International Conference on Environmental and Computer Science*, 2009. ICECS '09., p.19-23, 28-30 Dec. 2009 doi: 10.1109/ICECS.2009.43
- [3] BASSIL, Youssef. A simulation model for the waterfall software development life cycle. *arXiv preprint arXiv:1205.6904*, 2012.
- [4] ELSHANDIDY, Heba et MAZEN, Sherif. Agile and Traditional Requirements Engineering: A Survey. *International Journal of Scientific & Engineering Research*, 2013, vol. 9.
- [5] SHARMA, Sheetal, SARKAR, Darothi, et GUPTA, Divya. Agile processes and methodologies: A conceptual study. *International journal on computer science and Engineering*, 2012, vol. 4, no 5, p. 892-898.
- [6] MOHAMMED, Haifa et RAUF, Abdul. Agile Project Management: Brief Review. *Lecture Notes on Software Engineering*, 2015, vol. 3, no 3, p. 225.
- [7] KUMBHAR, Vidya et SHROTRIYA, Kanchan. Applying Scrum Practices to Improvise Placement Process of Prospective Employees. *Proceedings of the 5th national conference*.
- [8] CHO, Juyun, HUFF, R., et OLSEN, David. Management guidelines for scrum agile software development process. *Issues in Information Systems*, 2011, vol. 12, no 1, p. 213-223.
- [9] KHRAMTCHENKO, Serguei. Comparing eXtreme Programming and Feature Driven Development in academic and regulated environments. *Feature Driven Development*, 2004.
- [10] GOYAL, Sadhna. Major Seminar On Feature Driven Development. 2008.